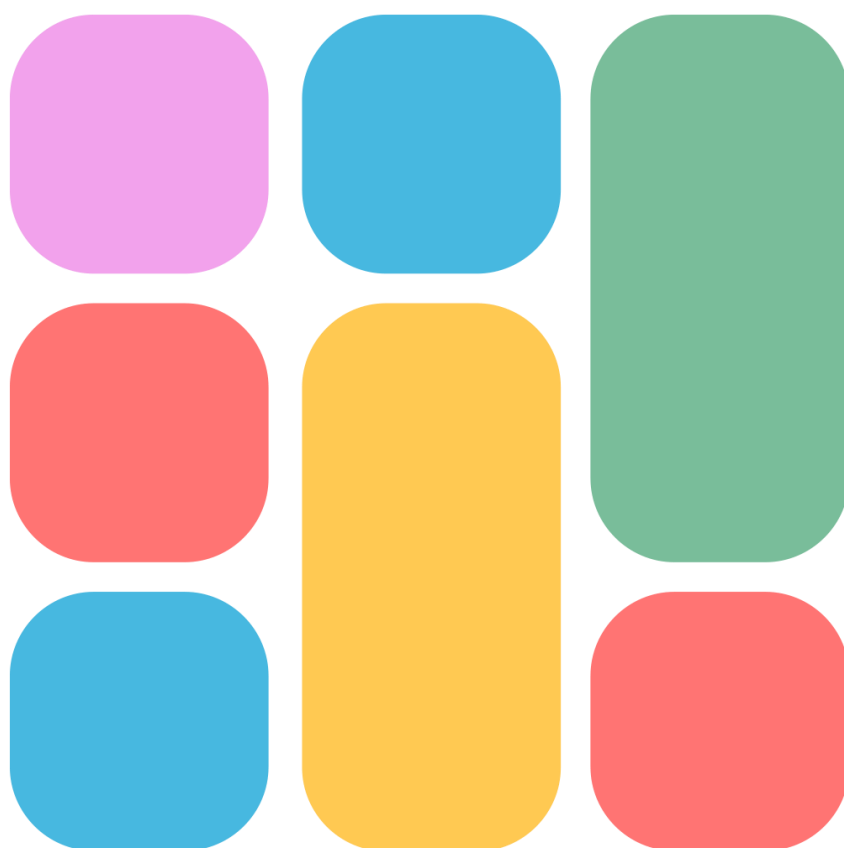




SÚNION

Generador d'horaris per a un centre escolar

Treball de Final de Grau
Juny 2016



Autor: Gerard Bayego
Ponent: Javier Béjar
Tutor: Tomàs Winand
Empresa: Escola Súnion
Especialitat: Computació
2n Quadrimestre 2015-16

Agraïments

Moltes gràcies a tots els que m'han donat suport i ajuda en el treball, especialment a Moisès Ruiz per la seva paciència i consells.

Resum

La resolució de problemes d'horaris és un àmbit llargament estudiat i investigat, considerant la seva importància en nombroses activitats i necessitats del món real. Moltes empreses necessiten solucions a aquest problema ja que si hiensem, els trens funcionen així, les plantes de producció, escoles, etc. Tot allò relacionat amb assignar uns certs recursos en el temps i optimitzar-los, pertany a aquesta família de problemes.

Aquest problema està sota el paraigua dels problemes NP-complert, és a dir que no es pot resoldre en temps polinòmic. Els problemes NP-complert, per poder resoldre'ls, s'han d'aplicar tècniques computacionals i d'intel·ligència artificial per poder arribar a la solució en un temps raonable, ja que és impossible provar totes les combinacions.

En aquest projecte ens trobem davant del problema de l'Escola Súnion, que necessita un software de generació d'horaris personalitzat per les seves necessitats específiques, ja que generen horaris setmanals segons les necessitats educatives del moment.

En aquest moment hi ha una persona que es dedica a jornada completa a dissenyar horaris a mà complint totes les restriccions necessàries. Amb el programa que es presenta en aquest document es pretén automatitzar aquest procés de confecció de l'horari.

Abstract

Problem resolution of timetables is an issue which is fully studied and researched, keeping in mind its importance in a large number of activities and needs in the real world. Many companies need solutions for this kind of problem, if we think about it, trains work this way and so do production lines, schools, etc. Everything which is related to certain time resources and its optimization is associated to this kind of problem.

This issue is included in the wide range of problems of the NP-complete, thus it can't be solved in polynomial time. Computational techniques and Artificial Intelligence are necessary to be able to solve NP-complete problems within a reasonable timeframe, since it's impossible to test all possible combinations.

In this project we are facing the problem of the Escola Súnion, which needs a software to generate personalized timetables for its special needs, since the school generates weekly timetables depending on the educational needs of each situation.

At this moment there is a full-time employee who manually designs timetables complying with all necessary restrictions. The programme that is presented in this document manages to automate the process of this timetable creation.

Índex

1 – Visió general	9
1.1 - Introducció	9
1.2 – Problema i actors	10
1.3 - Abast	11
2 - Estat de l'art	13
2.1 – Anàlisi de solucions existents	13
2.1.1 - Automated College Timetable Generator	13
2.1.2 – Automatic Time Table	14
2.2 – Anàlisi d'algorismes	15
2.2.1 – Backtracking	15
2.2.2 - Cerca local	16
2.2.3 - Satisfacció de restriccions	16
2.2.4 – SAT	16
3 – Desenvolupament	17
3.1 - Metodologia	17
3.1.1 - Anàlisi	17
3.1.2 – Disseny	18
3.1.3 – Codificació	18
3.1.4 - Proves unitàries	18
3.1.5 - Posada en producció	18
3.1.6 – Revisió de la metodologia	19
3.2 – Vista general de la solució	20
3.3 – Base de Dades	21
3.4 – Control d'execució	22
3.5 – Interfície d'entrada de dades	23
3.5.1 - Tecnologies	23
3.5.2 – Pàgines	24
3.5.3 – Classes	27
3.6 – Generador	28
3.6.1 - Tecnologies	28
3.6.2 – Funcionament general	28
3.6.3 - Algorisme	29
3.6.4 – Anàlisi de costos	30

3.6.5 – Classes.....	31
3.6.6 – Validació	32
4 – Implantació	33
4.1 – Migració	33
4.2 – Instal·lació del generador	33
4.3 – Formació	34
5 – Gestió del projecte	35
5.1 – Planificació	35
5.1.1 - 1r període – Anàlisi.....	35
5.1.2 - 2n període – Disseny	35
5.1.3 - 3r període – Codificació + Proves unitàries.....	35
5.1.4 - 4t període – Posada en producció.....	36
5.1.5 - 5è període – Documentació	36
5.1.6 – Possibles desviacions	36
5.1.7 – Canvis en la planificació inicial	36
5.2 – Costos	38
5.2.1 – Identificació dels costos.....	38
5.2.2 - Estimació dels costos.....	38
5.2.3 – Control de costos	40
5.2.4 – Desviació respecte al pressupost inicial	40
5.2.5 – Cost final	41
5.3 – Lleis i regulacions	41
5.4 - Coneixements d’assignatures cursades	41
5.5 - El projecte s’adequa a l’especialitat?	42
5.6 - Competències tècniques	42
6 - Sostenibilitat.....	45
6.1 – Matriu de sostenibilitat	45
6.2 – Dimensió Econòmica	45
6.3 – Dimensió Social.....	45
6.4 – Dimensió Ambiental	46
7 – Conclusions i futur	47
8 - Bibliografia.....	48
8.1 - Referències	48
8.2 - Altres.....	48

9 – Annex	49
9.1 – Manual de l'usuari	49
9.1 – Diagrama Base de Dades	51
9.1 – Gantt planificació inicial	52
9.2 – Gantt planificació modificada	53

1 – Visió general

1.1 - Introducció

A l'Escola Súnion (centre d'ESO i Batxillerat), com a part del sistema pedagògic, tenen un horari canviant, és a dir que a diferència dels centres escolars convencionals que a principi de curs preparen un horari setmanal fixe per a cada classe de cada curs, a Súnion cada setmana hi ha un horari diferent.

Com a particularitats afegides, a part del fet de ser canviant, podem trobar:

- Les classes poden tenir una durada variable segons les necessitats. L'horari segmenta els dies de la setmana en franges de 15 minuts, a partir d'aquí es poden fer classes on la duració sigui un múltiple d'aquest valor, tot i que habitualment les classes són de 45 minuts.
- No sempre en les activitats hi participen tots els alumnes d'una classe. El curs esta dividit en classes com a qualsevol centre escolar, però cada classe es subdivideix en 4 grups de 7-9 persones cada un (per exemple 1r d'ESO té 4 classes i 16 grups). Aquesta divisió fa que es puguin fer activitats amb qualsevol combinació de grups, des d'una activitat amb tots els alumnes (els 16 grups a la vegada), fins a activitats que fan de manera individual cada grup, passant per les classes normals que hi participen 4 grups, per tant, una classe.
- Les aules també van canviant, cosa que fa que els alumnes s'hagin de desplaçar per l'escola durant el dia. El canvi d'espais assignats pels alumnes i professors està determinat per les necessitats didàctiques de cada sessió, ja que es diferencien les classes teòriques, pràctiques, seminaris, treballs en grup, etc.

Per poder dur a terme aquest model d'horari, cal fer la planificació setmanal de manera manual, intentant complaure el màxim de peticions per part del professorat, les directrius determinades per cada àrea, curs i la direcció del centre; mantenint els requeriments oficials, és a dir nombre d'hores de cada assignatura que s'han de fer al final del curs.

El professorat pot fer tota mena de peticions. Veiem-ne alguns exemples:

- Demanar alguna classe de durada diferent a l'habitual a causa de les necessitats.
- Demanar una aula concreta per la necessitat de fer servir l'equipament que pugui tenir (aules amb ordinadors, aules amb projector, aules de plàstica, sala de projecció...).
- Demanar una classe amb agrupacions d'alumnes no habituals.
- Demanar temps lliure a una hora concreta com a conseqüència d'una reunió o altres possibles casuístiques.

Actualment, per dur a terme la confecció d'aquest horari, hi ha un treballador de l'empresa que es dedica a fer-los a mà, amb l'ajuda d'una eina visual tipus drag&drop on pot veure els diferents cursos i dies de la setmana per anar construint l'horari a

partir de totes les restriccions i peticions que té, intentant satisfer-ne el màxim nombre.

1.2 – Problema i actors

El projecte consisteix a fer una eina que generi horaris de forma semi-automàtica, és a dir que siguin els propis actors els que introdueixin les dades de les restriccions i peticions i que el programa ho resolgui amb un horari vàlid.

Les restriccions de l'horari són molt variades, però podríem segmentar-les en tres subgrups:

- **Restriccions de forma:** Totes les restriccions que tenen a veure amb el concepte que tenim d'horari (varien molt poc).
 - Dies lectius
 - Nombre de cursos i grups
 - Segmentació d'un dia en quarts d'hora
 - Els professors no poden tenir dues classes a la vegada
 - No pot haver-hi dues classes a la mateixa aula
 - ...
- **Restriccions de normativa:** Totes les restriccions que tenen a veure amb la planificació del curs (varien poc)
 - Cada curs té un conjunt d'assignatures a realitzar
 - Cada assignatura té un nombre de sessions setmanals determinat amb una durada determinada
 - Hi ha activitats que estan fixades en el temps
 - ...
- **Restriccions de demanda (peticions):** Totes les restriccions que tenen a veure amb la planificació del professorat de les seves activitats (varien molt)
 - Es necessita una aula de tipus concret per a una activitat
 - Es necessita una sessió de durada diferent de la predeterminada
 - Es necessita que certs professors tinguin un espai de temps reservat sense classe
 - Incidències tècniques que no permeten realitzar la classe en un espai concret.
 - ...

El programa, per poder fer la generació d'horaris, necessita la informació de les restriccions, per això part del projecte consisteix en la construcció d'una **interfície d'entrada d'informació**. En primer lloc, per les restriccions de forma i de normativa, serà una aplicació d'ús poc freqüent que hi tindrà accés la persona responsable de l'horari. En segon lloc, es farà una aplicació (probablement web) on el **professorat** podrà accedir amb la **seva clau** de la xarxa de Súnion, per **determinar les seves restriccions de demanda**.

Tanmateix, aquesta interfície ens ajuda a definir els actors implicats amb el producte final. Podem diferenciar fàcilment 2 tipus d'actors:

- **El supervisor/responsable:** encarregat d'entrar totes les restriccions de forma i normativa i donar per bons o no els horaris perquè puguin ser publicats oficialment.
- **Els professors:** cadascun d'ells pot entrar les seves peticions al sistema de manera que el programa les tingui en compte per al càlcul de l'horari.

Per tant, el projecte consistirà a fer la interfície d'entrada d'informació i el generador d'horaris que sigui capaç de resoldre els dos primers tipus de restriccions i de les restriccions de demanda les més comunes, tot preparant el programa perquè se'n puguin afegir fàcilment en un futur.

1.3 - Abast

Aquest projecte principalment pretén aconseguir fer un programa que pugui generar els horaris en un temps raonable i aconseguir la màxima expressivitat possible, per així simular al màxim la tasca de la persona que actualment s'encarrega de fer els horaris. En definitiva, reduir de forma considerable els recursos necessaris de la generació actual de l'horari.

Els tres principals problemes que apareixen són:

- **L'expressivitat:** cal aconseguir una expressivitat com més alta millor, és a dir, que el programa contempli com més casuístiques millor. D'aquesta manera es podria estalviar feina al responsable de l'horari en cas que la solució que retornés el programa necessités ser perfeccionada manualment abans de publicar-se. S'implementaran els dos primers tipus de restriccions i la major part possible de les de demanda. Tot plegat fa que els algorismes a fer servir es redueixin molt, principalment a codificació SAT o heurístics, tot i que es pot fer un sistema híbrid.
- **Presa de decisió:** hem de dissenyar un bon mètode de presa de decisions perquè el programa pugui escollir de la millor manera de solucionar un conflicte en el moment de generar l'horari. Per exemple si en el mateix moment dues classes necessiten la mateixa aula, i no hi ha cap altra manera de situar les classes, qui té preferència?
- **El temps:** cal aconseguir que el programa generi una solució en el menor temps possible. El problema és que com més augmentem l'expressivitat més augmenta el temps, i segons quins criteris de decisió es facin servir, també pot fer augmentar moltíssim el temps. A més s'ha de buscar algun mètode per poder arreglar els canvis d'última hora, ja que aquests es necessiten immediatament.

2 - Estat de l'art

Fixant-nos primerament en un àmbit més general, podem trobar molta documentació de teoria de planificació¹ i algorismes de IA¹ per resoldre problemes d'aquest tipus. De fet, després de fer la recerca veiem que podem fer servir diversos mètodes per abordar el problema, sigui amb mètodes de restriccions lògiques² o amb cerca local³.

En l'àmbit de la generació d'horaris hi ha moltíssima informació, ja que és un problema extensament estudiat. Molts sectors tenen les seves pròpies eines per l'organització d'elements en el temps: planificació de mitjans de transports, de centres educatius, amb finalitats hospitalàries.

Podem trobar programes ja creats que generen horaris però a tots els falta expressivitat per poder col·locar totes les possibles restriccions o situacions que es poden produir en l'horari de Súnion. I més enllà de la pròpia expressivitat, la finalitat d'aquestes eines s'allunyen molt de la que es vol aconseguir amb el generador automàtic per aquest centre.

En el sector amb el que es troba aquest projecte, tots els programes ja existents generen horaris seguint el típic model bàsic amb x classes, y professors-assignatura s'ha de generar unes possibles assignacions d'uns horaris sense solapaments.

2.1 – Anàlisi de solucions existents

2.1.1 - Automated College Timetable Generator

Centrant-nos amb un possible exemple, trobem Automated College Timetable Generator⁴, que és una eina web que et permet generar horaris a partir del mètode bàsic comentat anteriorment.

Aquest programa es segmenta a nivell d'interfície en els apartats següents:

- **Cursos:** Pots definir els cursos del centre escolar, tot donant-li un nom i dient quines assignatures fan i quin professor fa cada assignatura.
- **Professors:** Crees els professors, tot dient el nom del professor i un identificador.
- **Classes:** Crees les classes dins del curs.
- **Generar horari:** Genera l'horari a partir de les dades entrades.

Com podem veure aquest programa no satisfà les necessitats del nostre problema, donat que no et permet:

- Escollir la duració de les classes
- Canvis d'aula: consideren que un curs sempre esta en la mateixa aula.
- Assignatures diferents dins d'un mateix curs.
- Professors diferents segons la agrupació dins del curs
-

Automated Time-Table Generator

Add Course Add Teacher Add Class Room Create Time Table Logout

Add Course

Course Name :-

No Of Subjects :-

Subject 1 Name :-

Subject 1 Teacher :-

Priority :-

Subject 2 Name :-

Subject 2 Teacher :-

Priority :-

Subject 3 Name :-

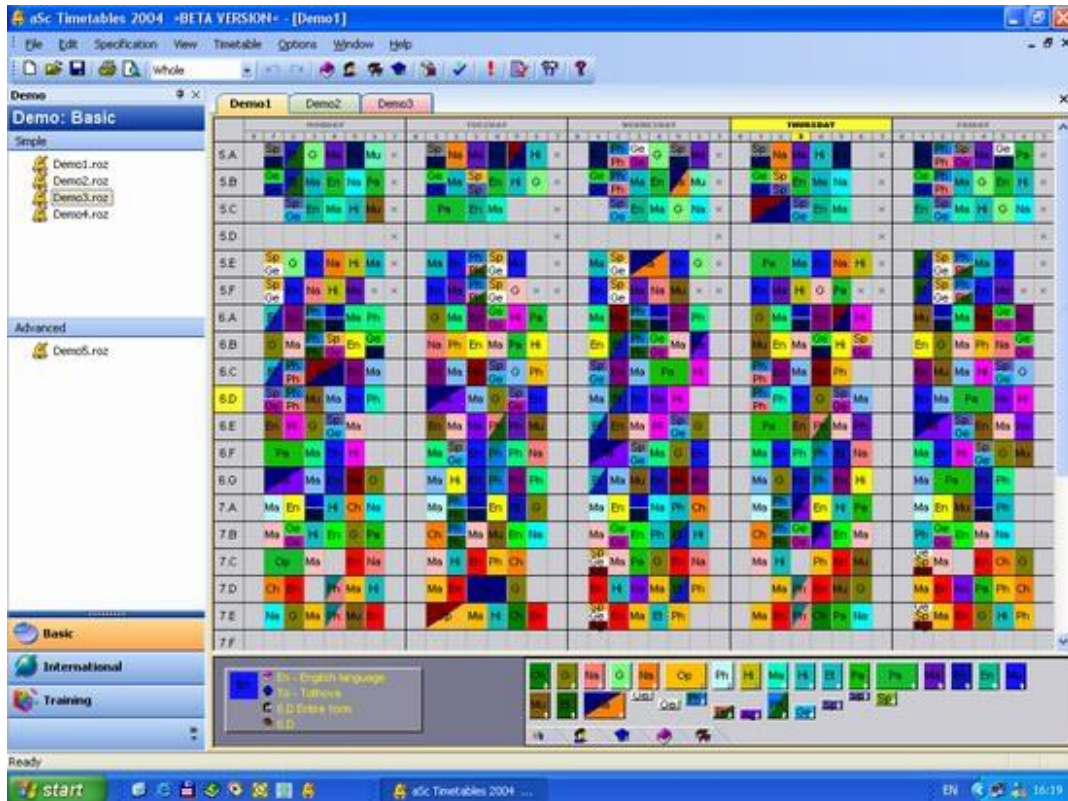
Subject 3 Teacher :-

2.1.2 – Automatic Time Table

Aquesta eina també funciona amb el mètode bàsic que hem comentat, tot i que és més expressiva a nivell de agrupacions possibles dins del curs, ja que et permet decidir quantes i quines son les agrupacions i definir assignatures diferents segons l'agrupació.

Com a funcionalitat extra té un sistema que et permet reubicar classes de forma més o menys simple en cas de malaltia d'un professor o alguna casuística que impedeixi realitzar alguna activitat.

De tota manera ens segueixen faltant possibilitats, sobretot dins del àmbit de la duració i posicionament de les classes, ja que tots aquests programes fan servir uns segments predefinits que no es poden moure i nosaltres necessitem la segmentació per segments de 15 minuts amb duració de classes variable.



2.2 – Anàlisi d'algorismes

Un cop analitzades les solucions existents, s'havien d'analitzar els possibles mètodes de resolució de problemes d'aquest tipus. Per començar es va fer una llista de possibles mètodes que ens podien ser útils, per després analitzar-los.

2.2.1 – Backtracking

El backtracking s'assembla a una cerca en profunditat dins d'un arbre, ja que es fa servir un arbre de possibilitats, que es va recorrent des de l'arrel fins a les fulles (que representen solucions possibles).

Aquest mètode el que es fa és generar una solució de manera incremental, és a dir, posant les peces d'una en una. Al fer-ho es construeix l'arbre de solucions parcials, on un cop complet, tindrem totes les possibles combinacions d'elements.

Òbviament no s'ha de generar tot l'arbre, sinó que l'algorisme para de provar combinacions en el moment que arriba en una fulla de l'arbre que és una solució vàlida.

Moltes vegades es fa servir una funció heurística per guiar la construcció de l'arbre, on la funció heurística ens indica, numèricament, de tots els camins possibles que tenim, quin és el més adequat provar.

2.2.2 - Cerca local

Aquest algorisme consisteix en, a partir d'una solució inicial, intentar-la millorar a partir d'aplicar operadors de canvi en la solució. Per poder fer servir aquest algorisme hem d'aconseguir, òbviament, generar una solució inicial de manera fàcil, per després poder millorar-la.

A més en aquest mètode es fa servir una funció heurística que ens informa de quina bonança té aquella solució representada normalment amb un número, per així poder-la compara amb les solucions veïnes i quedar-nos amb la millor.

Els operadors de canvi en la solució, busquen fer alguna modificació a la solució actual, de manera que es generen solucions semblants que poden tornar a ser avaluades amb la funció heurística i escollir-ne la millor.

Un exemple de cerca local podria ser el famós algorisme de hill climbing, que el que fa és buscar el millor canvi que pot fer a partir de la solució existent, en aquest cas buscant el màxim pendent a part dels resultats de l'heurístic.

Altres algorismes de cerca local podrien ser:

- **Simulated annealing**: que funciona a partir d'intentar simular el procés de refredament dels metalls.
- **Algorismes genètics**: que es basen en els mecanismes de selecció natural.

La cerca local no es pot assegurar que arribi a la millor solució possible. Això és causat pel fet que no podem visitar totes les solucions possibles, ja que l'espai de solucions és massa gran.

2.2.3 - Satisfacció de restriccions

La satisfacció de restriccions consisteix a fer un graf, amb el conjunt de restriccions que volem resoldre per, posteriorment, fer servir un backtracking cronològic per anar fent propagació en les restriccions i acabar solucionant el problema.

2.2.4 – SAT

Els problemes de SAT (satisfactibilitat booleana), són un tipus de problema on interessa saber si una expressió booleana amb variables, té una assignació de valors a les variables que fa que sigui satisfactible, és a dir que la fórmula s'avalui a vertader.

A partir d'aquestes fórmules de lògica proposicional, un SATsolver ens pot trobar els valors adequats per les variables de manera que satisfaci el sistema lògic. En el nostre cas es podria abordar el problema amb SAT fent una conversió de les restriccions i dades a fórmules lògiques, per poder enviar-ho a un SATsolver i amb el resultat transformar-lo a una solució de l'horari.

3 – Desenvolupament

3.1 - Metodologia

La complexitat del projecte recau en gran part en la definició d'un procés que compleixi totes les restriccions, casuístiques i condicions indicades anteriorment; però per arribar a aquest punt és necessari definir una metodologia de treball plantejada en diverses fases.

3.1.1 - Anàlisi

En primer lloc, és necessari l'anàlisi exhaustiu de la situació actual, conèixer els recursos humans i tecnològics utilitzats actualment per la tasca que es pretén millorar. Això implica l'estudi de les tecnologies utilitzades al centre per la generació i visualització de l' horari així com l'arquitectura informàtica de la que fa ús.

Els coneixements del treballador que actualment realitza la tasca manual de generació dels horaris, donarà una visió global del producte. Les plataformes, llenguatges i les bases de dades del generador i visors ajudarà a entendre el plantejament inicial amb el que es va realitzar l'horari manual. Molt probablement, les exigències del generador, obligarà a la millora de l'arquitectura informàtica que actualment s'inverteix per aquest fi.

Una vegada es coneix el context del projecte, es realitzarà l'anàlisi detallat de la problemàtica i abast del nou generador així com l'anàlisi d'especificacions i requeriments. Aquests requeriments proveiran d'unes especificacions determinades des de diverses fonts, com són el centre, els recursos i la legislació vigent en matèria educativa.

El centre, per la seva banda especifica les seves necessitats davant la proposta d'aquest nou producte, ja que és possible que a més d'aquesta generació automàtica, especifiqui noves funcionalitats. Un cop definides s'analitzarà la seva viabilitat tècnica i de costos per tal d'acabar perfilant amb l'empresa les especificacions de requeriments final.

Respecte als requeriments determinats per la legislació educativa, portarà a l'estudi de les normatives que deu complir el programa amb aspectes que el nou generador haurà de contemplar com són les hores lectives de cada assignatura oficial, les lectives màximes que poden haver cada dia, etc.

Un cop finalitzat l'estudi del sistema actual, fet l'anàlisi i l'especificació de requeriments, es realitzarà el disseny del projecte. Es definiran de forma conceptual els algorismes principals, l'arquitectura de base de dades, els diagrames de flux per tal de poder focalitzar posteriorment tots els esforços en la codificació del generador i l'entorn web per l'entrada de dades.

3.1.2 – Disseny

L'objectiu d'aquesta fase és principalment posar sobre paper tot el que s'ha recollit a l'anàlisi, per poder passar posteriorment a la fase de codificació.

El disseny està diferenciat en dues àrees: L'entorn web d'introducció de restriccions per part dels actors i de l'administrador per una banda, i el disseny del generador automàtic de l'horari canviant.

En l'àrea de l'entorn web, el disseny s'enfoca molt més a la part funcional, aconseguir que tots els requeriments que han aparegut a l'anàlisi es puguin dur a terme de manera fàcil i ràpida. La part de codificació d'aquesta àrea no és complexa.

En canvi a l'àrea del generador, el disseny és completament d'estructura de dades i algorismes per aconseguir que el programa faci l'esperat en un temps raonable. Per això s'ha de profunditzar molt a l'hora de buscar l'estat de l'art en aquest àmbit.

3.1.3 – Codificació

En aquesta tercera fase, la de codificació, s'ha de fer la implementació dels dissenys proposats. És molt possible que en el procés de codificació apareguin alguns punts no tractats explícitament en el disseny que s'hauran de resoldre, sobretot en l'àrea del generador i és el punt on apareixen els problemes. Per tant aquesta fase està molt relacionada amb l'anterior.

Per fer la codificació es farà servir un mètode incremental, de manera que primer es programarà una versió bàsica funcional i poc a poc s'aniran augmentant les funcionalitats fins arribar a la versió final.

3.1.4 - Proves unitàries

Aquesta fase té molta relació amb la fase de codificació ja que s'aniran fent proves a cada versió del programa des de la més bàsica fins la més complexa, d'aquesta manera es podrà assegurar el correcte funcionament del programa i serà més fàcil de trobar els problemes que puguin sorgir.

També ens permetrà comparar els temps d'execució segons la complexitat del problema i el nombre de funcionalitats. Per poder fer-ho s'hauran de dissenyar a mà jocs de proves que modifiquin els paràmetres i la complexitat, amb les seves respectives solucions. En el cas del problema dels horaris les solucions no tenen perquè ser úniques, poden haver-hi moltes solucions, per tant en els jocs de proves s'hauran de buscar casos extrems per assegurar el bon funcionament.

3.1.5 - Posada en producció

Un cop acabat el desenvolupament, s'ha de posar en producció l'eina. Per posar-la en producció s'haurà de fer formació als actors implicats, després d'haver col·locat el programa en l'entorn definitiu i fer les proves pertinents.

3.1.6 – Revisió de la metodologia

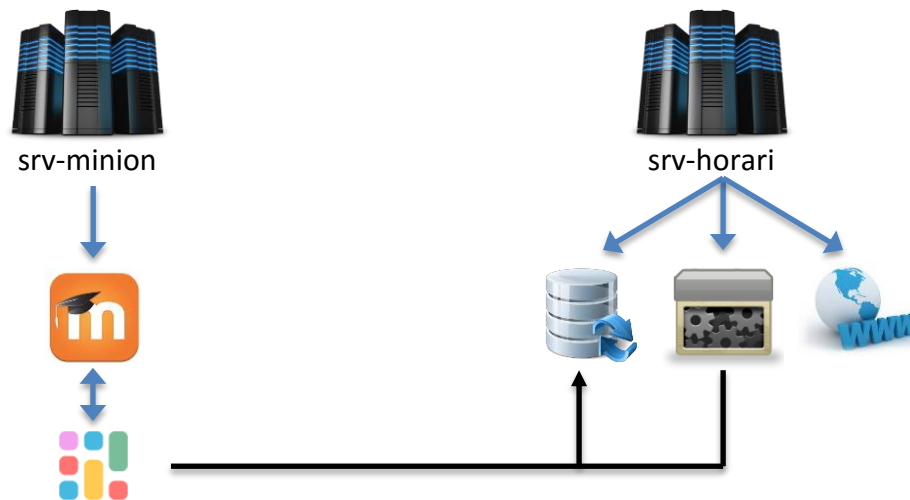
A mig projecte és un bon moment per fer una revisió de la metodologia, si està funcionant correctament i si s'ha de canviar alguna etapa. Podem dir que no necessitem fer-hi cap canvi ja que està funcionant tot correctament, de fet la metodologia incremental en la part de codificació i proves unitàries han afavorit al canvi de planificació que veurem més endavant, ajudant al correcte desenvolupament del projecte.

3.2 – Vista general de la solució

Tal com ja hem esmentat, la solució consta principalment de dues parts: la interfície d'entrada de dades i el generador d'horaris. Però per fer funcionar aquests elements tenim diversos elements d'infraestructura que s'han de comentar per fer-nos una idea general del programa.

Tot el sistema se sosté sobre dos servidors:

- **srv-minion:** Aquest és un servidor web, on s'allotja el Moodle de l'escola, anomenat intranet. S'union té un ERP de gestió escolar propi, que esta integrat amb Moodle i que està segmentat en mòduls, segons el contingut que gestionen. La interfície d'entrada de dades del TFG, s'ha programat com a un mòdul d'aquest ERP així doncs funciona integrat dins del Moodle del centre, tot i que les consultes de base de dades les fa contra la base dades del horari que està en el següent servidor.
- **srv-horari:** Aquest és un servidor web i de base de dades, on hi ha allotjat tot el sistema de l'horari i la respectiva base de dades. En aquest servidor se li ha afegit el generador d'horaris del TFG, que es controla a partir d'un cron que accedeix a la base de dades.



3.3 – Base de Dades

La base de dades es troba al srv-horari sustentat en MySQL. Aquesta base de dades ja té una estructura de taules que serveixen per fer funcionar l'horari amb el mode actual, és a dir, amb un programa que permet fer l'horari a mà i els corresponents mètodes de visualització.

Atès que el nou generador d'horaris no ha d'afectar el sistema actual, és a dir, que ha de seguir funcionant el mètode actual, els canvis que farem a la BBDD seran afegir taules, en cap cas canviarem l'estructura de les taules existents.

Actualment hi ha taules on es guarden les aules existents, els cursos, els departaments (arees), les assignatures, professors i la informació de l'horari de la setmana actual, on s'emmagatzemen, les anomenades targetes, que simbolitzen cadascuna de les activitats escolars, amb l'assignatura, professors i aula corresponents. Tota aquesta informació es copia a unes taules d'històric per poder analitzar-ne les dades posteriorment, per aquesta raó existeix una taula anomenada promocions, lloc on es guarden els identificadors del curs escolar, per poder segmentar totes les dades de la BBDD per curs escolar.

Per fer funcionar el nostre programa afegirem 13 taules noves:

- **Franges:** En aquesta taula guardem l'horari de cada curs de l'escola, és a dir, quins dies tenen classe i de quina a quina hora.
- **Setmanes:** En aquesta taula es defineixen les setmanes, és a dir, un nom i relacionar cada dia de la setmana amb un dia del calendari.
- **Indisponibilitat de professors:** Tenim dues taules on es guarda la informació de la indisponibilitat dels professors, per poder bloquejar l'assignació de classes en certs períodes de temps.
- **Sessions:** Consisteix en 6 taules que contenen la informació sobre les sessions predefinides que s'han de fer setmanalment, i les sessions que es faran a cada setmana que s'hagi creat a la taula de Setmanes. Cadascuna de les sessions té la informació associada dels professors que realitzen aquesta activitat i la informació de les aules que demanen per fer l'activitat.
- **Execucions:** En aquesta taula es guarda la informació bàsica d'una execució, és a dir, el id de la setmana que volem calcular i els timestamp d'inici i fi d'execució.
- **Anys:** En aquesta taula es guarda una relació entre les promocions que hem comentat anteriorment i el curs escolar del ERP de l'escola.
- **Moodle-prof:** En aquesta taula es guarda la relació entre el id de usuari a Moodle i el id a la taula de professors, d'aquesta manera podem segmentar la informació que mostrem segons l'usuari que accedeix a la app.

A l'annex podem trobar el diagrama de les taules de la BBDD.

3.4 – Control d'execució

Per poder gestionar la relació entre la interfície de dades i el programa generador, tenim un dimoni que s'ocupa d'aquesta tasca. Aquest dimoni s'executa un cop per minut i comprova si hi ha algun nou registre a la taula d'execucions a la BBDD.

Si troba un nou registre a aquesta taula, procedeix a:

- Guardar a la BBDD l'hora d'inici del programa.
- Generar un fitxer de text pla amb les dades necessàries per fer l'execució del generador, segons la setmana que s'ha seleccionat en crear l'execució.
- Executar el programa en segon pla amb aquesta entrada i espera a que acabi.
- Insertar a la BBDD l'hora de finalització del generador.
- Convertir el fitxer de sortida del generador a un fitxer que es pugui descarregar des de la interfície i que sigui compatible amb l'editor de l'horari.

Amb aquest sistema aconseguim una integració simple entre diferents servidors i llenguatges de programació diferents, recordem que la interfície és a un servidor separat de la BBDD i el generador.

3.5 – Interfície d'entrada de dades

L'entrada de dades consisteix a fer una web on els professors puguin entrar per posar-hi les seves restriccions i el gestor del horari configurar les dades bàsiques i controlar les execucions del generador.

A l'Escola Súnion estan en fase de desenvolupament d'un ERP intern, integrat amb Moodle, una plataforma de gestió de continguts educatius (campus digital). Aquesta plataforma permet instal·lar extensions gràcies al sistema modular que fan servir.

La interfície d'entrada de dades, per tant, l'hem integrat amb aquest gestor, que ens ha proporcionat una certa base en el funcionament, la gestió d'usuaris i els permisos d'accés.

Primerament es va fer tota la part nova de la base de dades que ja hem comentat anteriorment i un cop implementada, s'han construït les parts del gestor.

3.5.1 - Tecnologies

Tot aquest sistema funciona s'ha implementat usant les tecnologies PHP i MySQL a costat servidor i HTML5, CSS3 i Javascript a costat usuari.

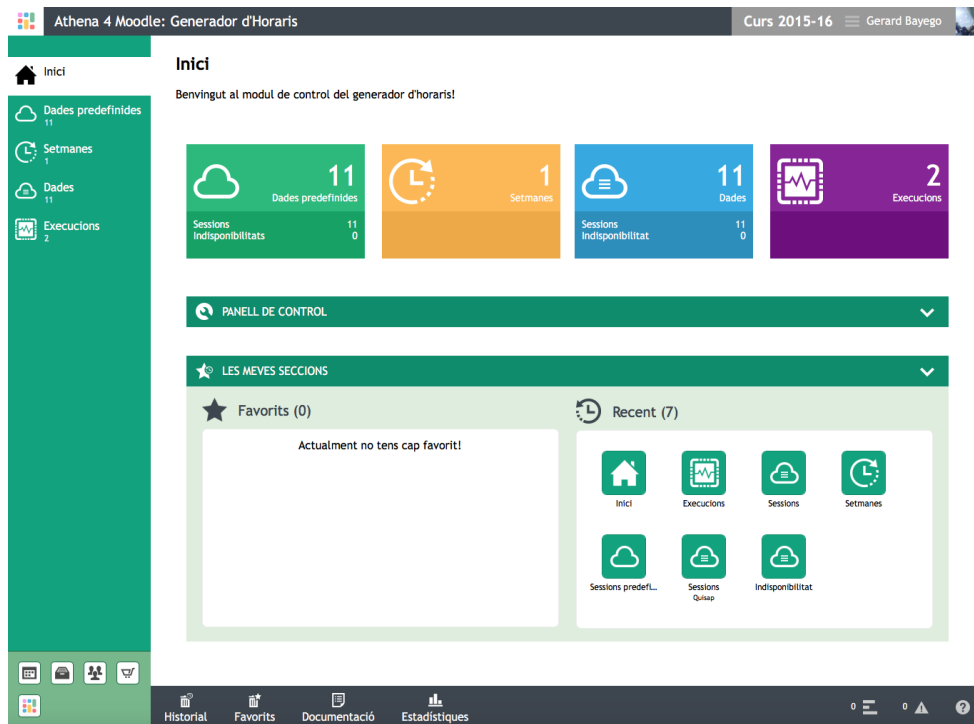
- **PHP:** És un llenguatge de programació molt estès a nivell de programació web, de fet tota la plataforma Moodle està implementada sobre PHP. És un llenguatge interpretat, imperatiu, orientat a objectes, amb tipatge dinàmic. Pot recordar a Java o C++ ja que son les principals influències que té.
- **MySQL:** És una famosa base de dades relacional mantinguda per Oracle. És software lliure i com a opció a destacar és que el llenguatge PHP té les funcions per accedir-hi en totes les seves versions, per tant tenim compatibilitat màxima. Per fer consultes a la base de dades es fa servir SQL.
- **HTML5:** És la mes recent versió del llenguatge HTML, és un llenguatge en forma de XML que és l'estàndard web actual, tots els exploradors web funcionen i llegeixen documents d'aquest tipus. En aquesta última versió s'han afegit moltes opcions d'usabilitat i dinamisme al llenguatge per poder facilitar al màxim la feina del programador, que abans s'havien de suplir aquestes mancances amb Javascript pur. Una de les grans novetats és la inclusió del canvas que està substituint el Java i el Flash.
- **CSS3:** És la mes recent versió del llenguatge CSS que serveix per donar estil als documents HTML, és una informació associada als tags del XML que conforma el document HTML, de manera que quan obrim una web es vegi "maca".
- **Javascript:** És un llenguatge de programació que s'executa l'explorador de l'usuari, serveix per aconseguir efectes visuals i dinamisme de la web. Ben implementat millora moltíssim la usabilitat de les webs. És un llenguatge interpretat, imperatiu, orientat a prototipus, funcional i amb tipatge dinàmic.

3.5.2 – Pàgines

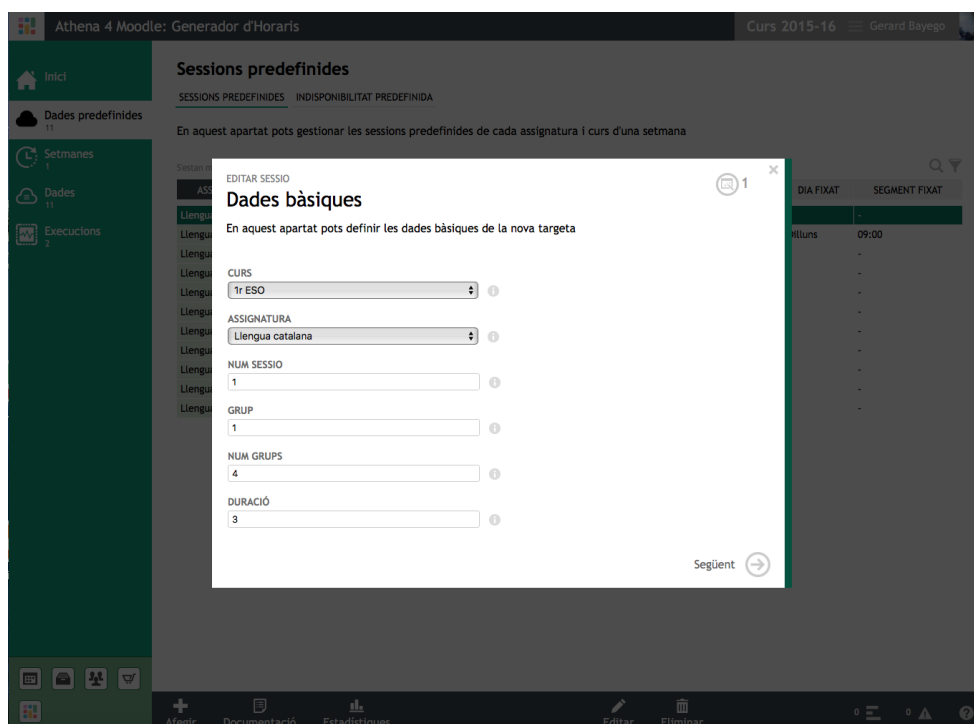
La interfície d'entrada de dades se segmenta en 5 parts: inici, dades predefinides, setmanes, dades i execucions.

Totes les parts tenen el mateix format:

- Informació (centre)
- Barra d'accions inferior
 - Globals (esquerra)
 - Associades amb la selecció que tinguem feta (centre)



Si fem click a una opció de creació/edició d'informació s'obrirà un assistent per poder dur a terme l'operació.



3.5.2.1 – Inici

Resum de dades informatives del mòdul, és únicament de visualització i d'accés ràpid als favorits i historial. En aquest cas mostrem la informació global de cada una de les altres 4 seccions.

3.5.2.2 – Dades predefinides

En aquest apartat es defineixen les dades predefinides, és a dir el nombre de classes que fa cada curs de cada assignatura, la durada habitual de cada una de les classes, els professors assignats a fer la classe, les aules preferides per fer-la, etc.

De la mateixa manera es poden gestionar les restriccions horàries habituals de professors, per exemple els professors que estan contractats a mitja jornada, tenen com a no disponibles les tardes de tota la setmana.

Athena 4 Moodle: Generador d'Horaris

Curs 2015-16 Gerard Bayego

Inici

Dades predefinides 11

Setmanes 1

Dades 11

Execucions 2

Afegir

Documentació

Estadístiques

Sessions predefinides

SESSIONS PREDEFINIDES INDISPONIBILITAT PREDEFINIDA

En aquest apartat pots gestionar les sessions predefinides de cada assignatura i curs d'una setmana

S'estan mostrant 11 de 11 entrades i en tens 0 seleccionades

ASSIGNATURA +1	CURS +2	GRUP +3	SESIÓ +4	DURACIÓ	NUM GRUPS	AULES	PROFESSORS	DIA FIXAT	SEGMENT FIXAT
Llengua catalana	1r ESO	1	1	3	4	-	ALI	-	-
Llengua catalana	1r ESO	1	2	3	4	-	ALI	Dilluns	09:00
Llengua catalana	1r ESO	1	3	3	4	-	ALI	-	-
Llengua catalana	1r ESO	13	1	3	4	-	ALI	-	-
Llengua catalana	1r ESO	13	2	3	4	-	ALI	-	-
Llengua catalana	1r ESO	13	3	3	4	-	ALI	-	-
Llengua catalana	1r ESO	5	1	3	4	-	ALI	-	-
Llengua catalana	1r ESO	5	2	3	4	-	ALI	-	-
Llengua catalana	1r ESO	5	3	3	4	-	ALI	-	-
Llengua catalana	1r ESO	9	1	3	4	-	ALI	-	-
Llengua catalana	1r ESO	9	2	3	4	-	ALI	-	-

3.5.2.3 – Setmanes

En aquest apartat es poden definir les setmanes per poder-hi treballar més tard. Per crear una setmana s'ha de definir el nom identificatiu, i dir de dilluns a divendres quins dies són lectius i quins no, tot relacionant el dia de la setmana amb el dia del calendari.

En el moment de crear una nova setmana, es copien totes les dades predefinides a la setmana en qüestió, de manera que les puguem editar de manera aïllada segons les necessitats d'aquella setmana.

3.5.2.4 – Dades

En aquesta secció primerament hem d'escollir una setmana al panell lateral que s'obre en entrar. Un cop escollida la setmana trobem exactament la mateixa interfície que a dades predefinides, però estem editant les dades de la setmana seleccionada.

Tal com ja hem comentat, en aquesta secció trobarem, inicialment, les dades predefinides, per així poder construir la setmana en qüestió a partir de les dades típiques.

Aquest és el lloc indicat per si un professor té una reunió un dia concret, que pot demanar la indisponibilitat d'una franja horària concreta d'aquella setmana o si es necessita fer una classe de durada diferent a l'habitual es pot fer el canvi que només afectarà la setmana seleccionada.

3.5.2.5 – Execucions

En aquest apartat podem crear noves execucions del generador. Per fer-ho ens demana de quina setmana volem fer el càlcul i quin temps màxim volem per fer l'execució.

Un cop creada l'execució veurem que no té data d'inici ni de finalització, ja que hem d'esperar que el dimoni del servidor de l'horari s'executi i comenci a fer la feina. Un cop el dimoni s'ha iniciat, apareixerà l'hora d'inici, i quan finalitzi apareixerà l'hora de finalització.

Un cop acabada l'execució podrem fer click al botó descarregar el fitxer de la solució, que podem obrir amb el programa d'edició ja existent. Des d'aquest programa podrem visualitzar la solució i fer-hi els canvis d'última hora, per, en últim lloc, publicar-lo com a horari definitiu.

Athena 4 Moodle: Generador d'Horaris Curs 2015-16 Gerard Bayego

Execucions

En aquest partat pots gestionar les execucions del algorisme generador d'horaris

S'estan mostrant: 1 de 1 entrades i en tens 0 seleccionades

IDENTIFICADOR ↓	SETMANA	HORA INICI	HORA FI	MAX TEMPS D'EXECUCIÓ
3	2 Maig	07/06/2016 17:10:00	07/06/2016 17:10:01	300

Afegir Documentació Estadístiques

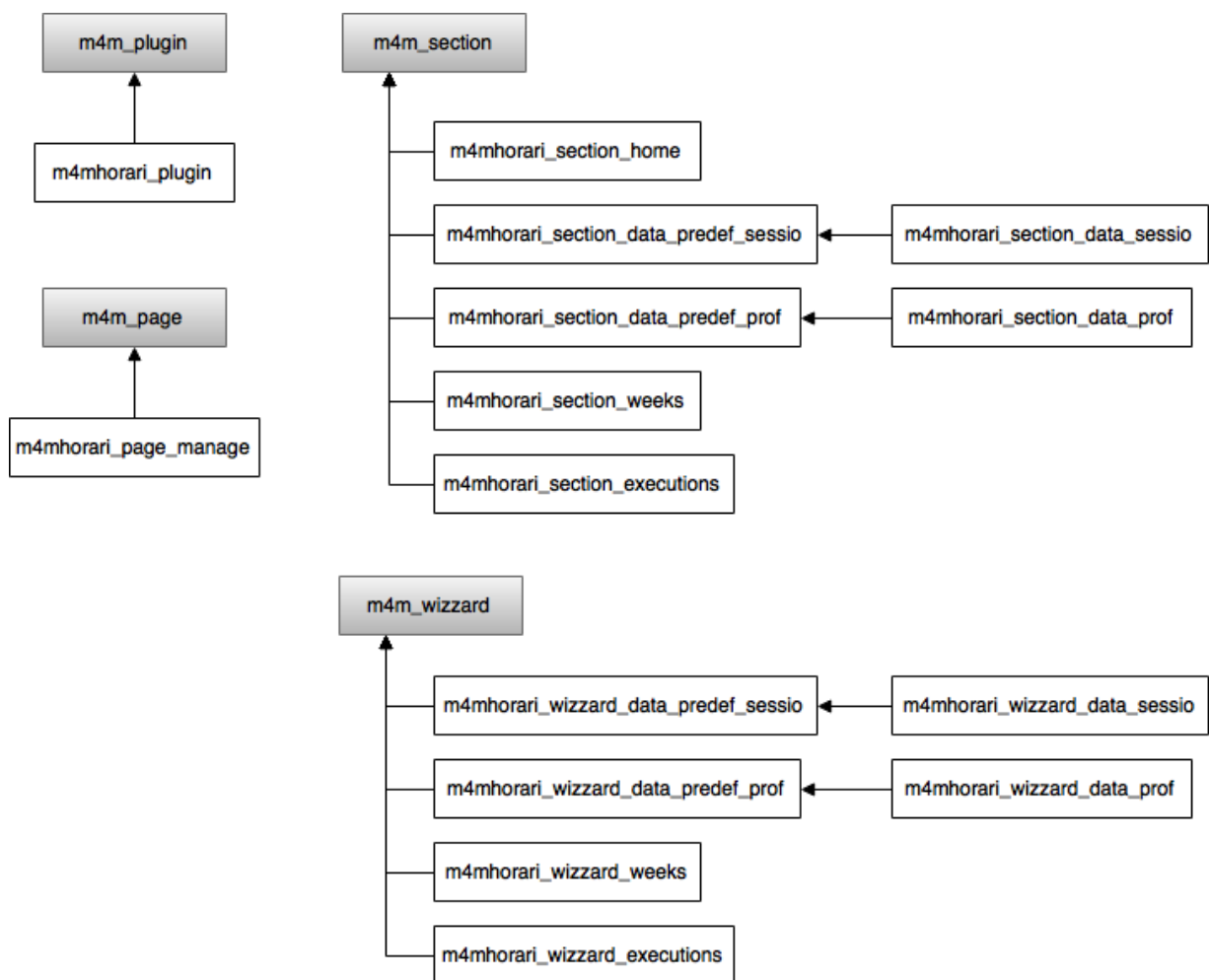
3.5.3 – Classes

Les classes dins de la interfície van directament relacionades amb les pàgines explicades anteriorment. Per cada pàgina trobem una classe que hereta de la classe abstracte m4m_section així com amb els assistents d'edició de dades, que hereten de m4m_wizzard.

A part, trobem dues classes especials que són, m4m_page i m4m_plugin que defineixen el contingut del mòdul, menú, etc... i que són accessibles des de qualsevol altra classe ja que s'inicialitzen com a objectes globals.

Podem veure que hi ha unes excepcions que són les classes de les seccions i wizards de dades, que hereten de dades predefinides, ja que com s'ha comentat ja fan servir exactament la mateixa interfície, només s'han de canviar les taules d'obtenció de dades.

Totes les classes marcades en gris pertanyen al core del ERP de Súnion, i són totes classes abstractes.



3.6 – Generador

El generador d'horaris, que és el nucli del treball, és on radica la part complexa en tecnologies i procediments i la part que a partir de les dades inserides a la interfície, s'ocuparà de processar i donar una solució vàlida a l'horari de la setmana demanada.

3.6.1 - Tecnologies

Primerament s'havia de decidir amb quin llenguatge de programació es faria el desenvolupament, pensant sobretot en el rendiment i estabilitat del programa, tot i que també influïa en la decisió els entorns de programació i debugging disponibles per a cada llenguatge.

Després de descartar llenguatges amb rendiment baix, es va acabar amb una decisió entre C++ o Java. Per comoditat es va escollir C++ ja que l'entorn de programació de Visual Studio 2015 ajuda molt al programador a l'hora de buscar errors. Té comprovació de codi en temps real i eines de debug molt potents, entre les quals profiling temporal de les funcions en temps d'execució, així com tot el sistema típic de breakpoints associats a condicions.

El llenguatge C++ és un llenguatge compilat, imperatiu, orientat a objectes, de tipat estàtic. En ser compilat té molt bon rendiment i té implementada una bona gestió de memòria. A part, el compilador de C++ té la possibilitat d'aplicar diversos nivells d'optimització de manera que els codis tindran un rendiment realment bó.

3.6.2 – Funcionament general

El generador se segmenta en 5 passos diferenciats:

- **Lectura de dades:** En aquest pas llegeix totes les dades necessàries i inicialitza les classes necessàries guardant els objectes corresponents.
- **Assignació de sessions fixades:** Inicialitzem una solució buida i li assignem les sessions que s'havien definit amb un punt del temps concret.
- **Execució de l'algorisme:** Executem l'algorisme per col·locar totes les altres sessions.
- **Assignació d'aules:** Un cop l'algorisme genera una solució es fa un repartiment aleatori de les aules no assignades a cada una de les targetes per així completar la solució.
- **Escriure solució:** Per últim, òbviament escrivim la solució per poder-la fer servir posteriorment.

3.6.3 - Algorisme

A partir de cada mètode que s'ha vist a l'estat de l'art, es va dissenyar una fusió de tots ells, per aconseguir un programa que ens servís pel nostre problema. La idea era fer un programa de satisfacció de restriccions a partir d'un backtracking amb heurístic, que tingués una cota de control de la qualitat de les solucions parcials acceptables (aquesta cota va variant amb el temps).

El nostre backtracking consisteix en anar afegint targetes de l'horari a partir d'un horari buit. Cada vegada que s'insereix una targeta, es proven totes les possibles posicions dins de l'horari i s'escullen les 5 millors segons l'heurístic, d'aquestes 5 se'n escull una aleatòriament, que és la que continua amb l'execució d'aquesta manera ja no estem fent l'arbre de possibilitats complert, sinó que esporguem branques que l'heurístic pronostica que no formaran part de la solució.

La cota de control limita les solucions parcials que estem generant, de manera que si la solució és més dolenta del permès, es guarda en una cua amb prioritats, i es continua la execució amb la millor solució parcial de la cua. Aquesta cota de control, es regeix per funció exponencial, segons el temps d'execució, al principi només accepta solucions amb cost molt baix i poc a poc es va acostant al temps màxim d'execució definit, va augmentant progressivament. Amb aquest comportament el que estem simulant és una cerca en amplada a l'inici de l'execució, és a dir que obre molts possibles camins i poc a poc va cercant en profunditat.

Cal recordar que gràcies a la cua de prioritats sempre estarem treballant sobre la millor solució parcial que tinguem fins el moment, encara que la cota de control sigui molt alta.

L'heurístic consta de dues parts, la funció predictora $h(n)$ i la funció de cost real $g(n)$. La funció predictora de cost diu quin és el cost previst per arribar a una solució vàlida, per tant, va disminuint amb l'avanç de la solució. La funció de cost real fa una valoració del cost que té actualment la solució, en el nostre cas segons el nombre de restriccions que està violant fa que el valor augmenti.

Tal com havíem comentat hi ha diferents tipus de restriccions, les de forma i normativa, són restriccions fortes per tant si s'intenten violar la solució s'invalida automàticament. En canvi, les de demanda són fluïxes és a dir que fan augmentar el valor de $g(n)$ però no n'invaliden les solucions. L'única excepció en aquestes regles és que els professors se suposa que no poden tenir dues classes a la vegada, a Súnion a vegades es dona aquest cas, són casos controlats, per tant, s'ha col·locat com a una restricció fluïxa i no invalidant com hauria de ser.

A part, per fer que el programa funcioni més ràpid, es fa servir una estratègia greedy que es fa servir en SAT solvers, que és ordenar les dades d'entrada segons el nombre de restriccions associades, per així aconseguir que el backtracking trobi com abans possible conflictes i elimini branques no possibles.

3.6.4 – Anàlisi de costos

El generador de l'horari per resoldre el problema té, com ja hem vist, varies fases, la fase que realment gasta tot el temps que tenim disponible és el Branch, per aquesta raó analitzarem les parts crítiques d'aquest procés.

Com ja hem vist amb anterioritat, el Branch es basa principalment en anar construint una solució afegint targetes. Cada vegada que n'afegeix una, el que fa és copiar la solució actual N vegades, per poder provar les N posicions d'una targeta, actualitzar el cost a cada una d'elles i escollir-ne una. Per tant veiem que els dos punts més crítics són la còpia d'una solució i el càlcul del nou cost.

3.6.4.1 - Còpia de la solució

La classe solució, te quatre atributs públics, dos enters i dos booleans, i una matriu privada. El cost de fer la còpia dels primers quatre elements és $O(1)$, en canvi el de la matriu depèn de la seva mida.

Per poder-ho analitzar correctament hem de analitzar les mides internes de la matriu, que en el nostre cas te 5 dimensions:

- Cursos (c): Els cursos que té l'escola, que són 6.
- Grups (g): Els grups de cada un dels cursos, als 4 cursos d'ESO son 16 i als dos de BATX 22.
- Dies (d): Els dies de la setmana, com a màxim i habitualment són 5.
- Franges (f) i Segments (s): Les franges representen el matí i la tarda per tant són 2 i els segments són els segments de 15 minuts de cada franja (20 al matí i 9 a la tarda). Per fer els números més fàcils introduïm la variable fs que son els 29 segments d'un dia.

Per tant la mida de la matriu és $c \cdot g \cdot d \cdot fs$ és a dir que el cost de la còpia de la matriu es $O(c \cdot g \cdot d \cdot fs)$ que clarament coneixent les dimensions dels paràmetres fa que sigui una operació molt costosa i que no podem millorar.

3.6.4.2 – Càlcul del nou cost

Per calcular el nou cost i fer que sigui el més eficient possible s'utilitza un mètode incremental, és a dir que es calcula el cost a partir de l'anterior.

En la funció heurística, que és l'encarregada de fer l'actualització del cost, primerament es fan una sèrie d'operacions de cost $O(1)$, que no afectaran al cost global, com podien ser disminuir en 1 el cost pel fet d'afegir una targeta o fer comprovacions de quan toca temps lliure.

Després trobem la part realment costosa que és la que comprova els conflictes de professors i aules. Per fer-ho s'ha d'iterar la solució, que el cost és $O(t \cdot c \cdot g \cdot p)$ pels conflictes de professors i $O(t \cdot c \cdot g \cdot a)$ pels conflictes d'aules, on t és la duració de la targeta en numero de segments, p el nombre de professors i a el nombre d'aules. Per tant el cost de l'heurístic és:

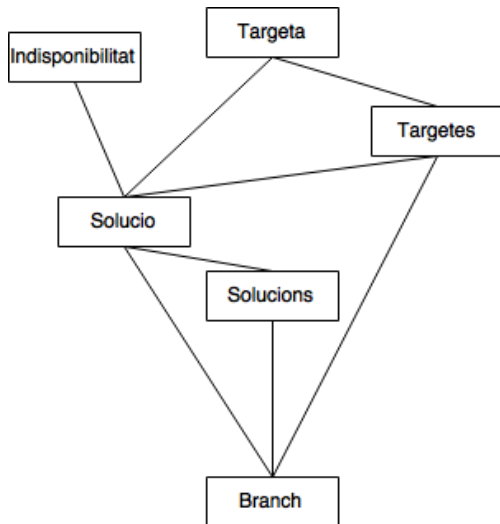
Si $p > a$ el cost és $O(t \cdot c \cdot g \cdot p)$

Si $p < a$ el cost és $O(t \cdot c \cdot g \cdot a)$

Al fer-ho incrementalment, aconseguim només haver d'iterar la mida de la targeta que estem afegint, que acostuma a ser 3, sinó hauríem d'iterar tots els segments d'una solució que acostumen a ser 5 dies · 29 segments per dia = 145 segments. Per tant aconseguim un 4800% de speedup.

3.6.5 – Classes

Anem a analitzar cada una de les classes del generador i les seves responsabilitats, a part en el diagrama podem veure quines són les dependències de cada classe.



Targeta: conté la informació bàsica d'una targeta, que és l'element que haurem d'assignar a la solució (id, curs, grup, número de grups, duració, professors i aules). A part també guarda la informació de la posició en cas de ser una targeta fixada.

Targetes: és un conjunt de targetes, està representat amb un map que associa id i objecte Targeta, d'aquesta manera la resta de l'algorisme pot accedir a la informació d'una targeta a través del seu id en temps log(n).

Indisponibilitat: és una classe que guarda la indisponibilitat dels professors, per accedir pots demanar-li per id de professor i un dia/franja concret.

Solució: Conté una possible solució del problema. Consisteix en una matriu on a cada posició podem trobar un -1 si és una posició buida o un enter major que 0 que representa el id de una targeta. Com a atributs importants te:

- Cost: Ens dona l'enter que ha calculat l'heurístic
- Finished: Booleà que ens indica si la solució és completa o no.
- Impossible: Booleà que ens indica si la solució és impossible.

També té un mètode públic que és el que fa servir el Branch per poder afegir una targeta a la solució. En executar-se aquest mètode, s'actualitzen les dades dels atributs comentats anteriorment i la representació interna de la solució.

Solucions: És el contenidor de solucions completes, on es van guardant mentre queda temps d'execució.

Branch: És la classe que executa l'algorisme del que hem parlat anteriorment.

3.6.6 – Validació

Per fer la validació del programa, s'han fet jocs de proves per comprovar els següents aspectes de cada classe del programa de manera aïllada:

- Casos extrems: Per assegurar que no es generen segmentation faults o problemes semblants. Aquesta prova un cop passada ja ens dona molta informació ja que si no estem accedint fora de la memòria, és que l'algorisme està funcionant dins dels paràmetres del disseny.
- Proves de funcionament: Amb exemples fets a mà s'ha executat de manera que generi els resultats que s'han previst.

S'ha de tenir en compte que hi ha algunes parts del programa que tenen punts d'aleatorietat, per poder comprovar el funcionament d'aquestes classes s'ha desactivat l'aleatori per un procediment conegut de manera que els jocs de proves fossin previsibles.

També s'ha de tenir en compte que si alguna classe té dependències d'una altra classe, s'han generat stubs amb dades conegudes per estar segurs de la influència de les altres classes.

Posteriorment de la validació aïllada s'han anat ajuntant les classes per ordre d'execució fins arribar a la solució final estable.

4 – Implantació

Un cop fet tot el desenvolupament en un servidor de proves, ha arribat l'hora de passar tot el sistema a productiu. Per fer-ho ens trobem amb el problema que el servidor actual de l'horari és poc potent i no pot aguantar la càrrega que suposa el programa de generació d'horaris, per tant haurem de fer la migració a un nou servidor.

4.1 – Migració

Primerament instal·lem el SO al nou servidor, on posteriorment hi instal·lem un servidor web + BBDD. Farem servir Apache com a servidor web, amb l'extensió de PHP que és amb el que estan programats els visors públic de l'horari. Com a base de dades, com ja hem comentat anteriorment, instal·larem un MySQL.

Un cop instal·lat tot el programari hem de copiar les configuracions del l'Apache, concretament del vhosts.conf, que conté els directoris i urls d'accés al servidor. A part de copiar la configuració copiarem també els arxius de codi font del visor de l'horari i farem un dump de la BBDD original per inserir-la al nou servidor.

Un cop fet això es canvia la direcció IP del servidor i s'apaga l'antic, de manera que el nou servidor pren el relleu.

4.2 – Instal·lació del generador

Ara que ja tenim el servidor definitiu, primerament hi crearem les noves taules de la base de dades buides, per tant farem una exportació de l'estructura des del servidor de desenvolupament i una importació en el de producció.

Posteriorment hi col·locarem el dimoni i crearem la "tasca programada" que s'encarrega d'executar-lo cada minut. Juntament amb el dimoni hem de col·locar el .exe del generador compilat i crear els directoris on es crearan els arxius d'entrada i sortida del programa.

Per anar acabant anirem al servidor de la intranet on instal·larem l'extensió de Moodle que conté la interfície d'entrada de dades.

Ja per últim hem d'assegurar-nos que tant la configuració del dimoni com la de la interfície d'entrada de dades apunten al servidor de base de dades de producció i no al de proves.

4.3 – Formació

L'última part de la implantació es realitzarà al inici del següent curs lectiu. Al tractar-se d'una escola no és possible realitzar la formació a final de curs pel volum de treball i tot el que implica finalitzar un curs escolar.

Es realitzaran sessions de formació tot just abans de començar el nou curs al professorat i a l'encarregat de l'horari per tal que tots els actors implicats d'alguna manera amb la interfície, coneguin les eines de configuració .

Quan finalitzin les formacions i utilitzin l'entorn d'entrada de dades, es farà un recull del feedback per part dels professors i l'encarregat de gestionar l'horari, per tal de realitzar un estudi de requisits posteriors a la primera fase i fer les millores pertinents.

Probablement, les millores estaran relacionades amb factors d'usabilitat o bugs menors que puguin aparèixer en l'ús intensiu de la interfície d'entrada de dades.

5 – Gestió del projecte

5.1 – Planificació

El projecte, en termes temporals, s'ha de realitzar completament entre el 1 de febrer i el 10 de juny, duració del contracte amb l'escola SÚNION, per tant en un període de 4 mesos i mig (18 setmanes). Tal com ja hem comentat a l'hora de especificar la metodologia, el projecte s'ha dividit en 5 fases: anàlisi, disseny, codificació, proves unitàries i posada en producció, per tant dividim el temps total en períodes.

5.1.1 - 1r període – Anàlisi

En aquesta fase s'ha de recollir tota la informació necessària, per tant la primera tasca és familiaritzar-se amb l'entorn del problema i la situació actual. Per fer-ho, s'ha d'analitzar el funcionament de l'horari a fons, i això significa parlar amb el responsable actual de fer l'horari manualment i veure com el fa.

Un cop recollida tota la informació, s'ha d'analitzar l'estat de l'art, referent als conceptes específics que s'han introduït. De fet aquesta fase correspon a la fita inicial de GEP.

La durada d'aquest període serà de 3 setmanes.

5.1.2 - 2n període – Disseny

En aquesta fase realitzarem el disseny tant intern com visual del programa, òbviament necessitem tenir tota la informació de com es fa l'horari. Principalment ho dividirem en dues parts. Primerament farem el disseny amb les dades que tenim, per poder anar a parlar amb el responsable de l'horari i presentar-li la proposta de la solució al problema.

Posteriorment la segona fase serà revisar i tancar el disseny segons les objeccions que hagi fet el client.

La durada d'aquest període serà d'un mes (4 setmanes).

5.1.3 - 3r període – Codificació + Proves unitàries

Donat que s'han de preparar les dues parts, la interfície per entrar dades i el programa generador d'horaris, ho farem separatament.

Primerament farem la interfície d'entrada de dades, ja que si no no tindrem dades com per poder començar a programar i provar el generador.

Un cop feta la interfície, podem començar a fer el generador d'horaris, que ho segmentarem en 2 fases, la bàsica, que generarà horaris sense les restriccions de demanda, i en una segona fase, afegirem les restriccions de demanda.

La durada d'aquest període serà de dos mesos (8 setmanes).

5.1.4 - 4t període – Posada en producció

En aquesta fase s'ha de posar en producció el nou software, que també es farà en dues fases.

En la primera hi haurà una comunicació constant amb el client per ensenyar-li el funcionament de la nova eina i si existeix algun defecte solucionar-lo ràpidament.

En la segona fase ja es farà servir el programa nou de manera definitiva, ja sense els possibles problemes que hagin pogut sorgir en l'anterior fase.

La durada d'aquest període serà de tres setmanes.

5.1.5 - 5è període – Documentació

Aquesta és l'última fase, es tracta de revisar la documentació prèviament feta, principalment tota la fase 1, i elaborar el document final amb tots els passos ben especificats i definits.

La durada d'aquest període serà d'una setmana.

5.1.6 – Possibles desviacions

Òbviament en tot projecte estem subjectes a possibles problemes i retards, per tant necessitem tenir un pla b per abordar aquestes desviacions del pla inicial.

En el cas que hi haguessin petites desviacions (1 – 2 dies) es podrien compensar amb hores extres, la qual cosa suposaria un cost addicional en el projecte, o es pot intentar fer ballar algun dia entre períodes.

En el cas que sigui una desviació major s'hauria de fer un replantejament de la planificació i probablement aplaçar la posada en producció. Per sort, en la posada en producció, tenim dues setmana extres de marge, abans que s'acabi el curs escolar.

5.1.7 – Canvis en la planificació inicial

Després d'una reunió amb el tutor del TFG i direcció de l'escola, es va decidir que el període de posada en producció, s'ajornaria al curs vinent, per tant fora del treball. Tot i així la part de fer proves amb el client, sí que es duu a terme, concretament l'última setmana del projecte (6 al 10 de juny).

Atès que inicialment s'havien previst 3 setmanes, tenim dues setmanes lliures que han servit per reubicar altres feines. Donat que el període de codificació del programa bàsic ha portat una setmana més del previst i el període de codificació de la web d'entrada de dades també, ja tenim les dues setmanes de desviació.

Com a altre canvi en la planificació, hem de remarcar que inicialment estava previst fer la interfície de dades, i posteriorment el generador en mode bàsic i avançat. Finalment s'ha canviat l'ordre per no haver de refer feina i s'ha comentat pel generador bàsic,

després l'entrada de dades i per últim en paral·lel el generador avançat i acabar l'entrada de dades.

Aquest canvi d'ordre com dèiem s'ha fet per estalviar feina, ja que si començàvem per la web no acabava de quedar clar amb plena exactitud quines dades necessitaria l'entrada del programa. Així en fer la versió bàsica ja s'ha vist l'estructura de tot el necessari i s'ha pogut fer la web sense problema.

5.2 – Costos

5.2.1 – Identificació dels costos

- **Recursos humans:** En aquest projecte es veuen implicats directament la figura del director de projecte, que principalment fa d'enllaç amb l'empresa, i la persona que realitza el TFG, que farà el rol de dissenyador i programador.
- **Hardware:** Com a hardware necessitem el PC on es durà a terme el desenvolupament i posteriorment un servidor on instal·lar el software.
- **Software:** Per la realització del projecte necessitarem software per dur a terme el desenvolupament i per fer la documentació.

5.2.2 - Estimació dels costos

5.2.2.1 - Recursos humans

- **Director de projecte:** S'estima que el director de projecte haurà de dedicar un total de 30h de feina dedicades a l'avaluació del TFG i a vetllar per una correcta finalització del projecte. El director del projecte és empleat de l'empresa i se li pagarà un plus ja que no és una de les seves feines habituals. Aquest extra serà de 30€/h per tant suposarà un cost de 900€.
- **Programador:** Tal com s'ha comentat abans el programador farà també les feines d'anàlisi i implementació ja que es tracta de un TFG. Per normativa el TFG ha de suposar una carrega lectiva de 375h, per tant, la durada del projecte es calcula en aquests termes. S'ha arribat a un acord amb l'empresa que el sou del programador serà de 10€/h. Aquesta figura és la que vam planificar el seu treball amb el diagrama de gantt, com en aquest cas el preu/h és el mateix sempre, podem dir que els costos directes segons les activitats del gantt es poden calcular amb les hores totals directament, per tant en total el cost serà de 3.750€.

Rol	Hores	Preu/h	Cost
Director	30h	30€/h	900€
Programador	375h	10€/h	3.750€
TOTAL	405h		4.650€

5.2.2.2 – Hardware

- **PC desenvolupament:** L'ordinador en qüestió ha costat 1.000€, considerant una amortització a 4 anys (48 mesos) i que es farà servir pel projecte durant 4,5 mesos, això suposa aproximadament un 10% de la vida útil, per tant el cost serà de 100€.
- **Servidor de càlcul:** Un cop acabat el desenvolupament s'ha de implementar el nou software a un servidor, el seu cost serà de 4.000€

Hardware	Preu	Amortitzar a	Cost
PC Desenvolupament	1.000€	4 anys	100€
Servidor	4.000€	-	4.000€
TOTAL	5.000€		4.100€

5.2.2.3 – Software

El software de pagament que necessitem el podem veure a la següent taula. Hi ha altre software que s'utilitzarà tipus compiladors, que no s'han inclòs a la llista ja que són software lliure o que per ser estudiant el podem obtenir gratuïtament i per tant no tenen un cost associat.

Producte	Cost
Office	150€
Dreamweaver	500€
SublimeText	70€
Windows server	800€
TOTAL	1.520€

5.2.2.4 – Despeses indirectes

Els projectes informàtics com tots els projectes tenen costos indirectes, en el nostre cas estan predominats per l'electricitat, considerem que del cost mensual d'electricitat de l'empresa que és uns 3.000€, el nostre projecte en fa servir un 0,5%

Concepte	Cost mensual	Cost
Electricitat	15€	70€
Menjar	90€	405€
Despeses administratives	-	1.200€
TOTAL		1.675€

5.2.2.4 – Imprevistos

Com a tot projecte poden sorgir imprevistos durant el procés, i ja es van analitzar els possibles inconvenients que podíem trobar. Un cop analitzats els possibles inconvenients s'ha de buscar un encaix econòmic de manera que tinguem una part pressupostada que potser no es fa servir, però que servirà de reserva per poder cobrir els imprevistos.

Per fer-ho farem servir el mètode del fons de contingència, per tant hem de definir un percentatge respecte el total del pressupost que servirà com a reserva per pagar els imprevistos si succeeixen.

Definirem el % de contingència al 15% per tant si el projecte té un cost total de 11.945€ la partida de contingència serà de 1.800€.

Part	Valor	%	Cost
Contingència	11.945€	15%	1.800€
TOTAL			1.800€

5.2.2.5 – Pressupost total

Per acabar veiem el pressupost resumit per parts.

Part	Cost
Recursos humans	4.650€
Hardware	4.100€
Software	1.520€
Despeses indirectes	1.675€
Imprevistos	1.800€
TOTAL	13.745€

5.2.3 – Control de costos

Per fer el control de costos hem de buscar algun indicador amb el qual puguem detectar desviacions durant la realització del projecte.

En el nostre cas és molt clar que l'indicador és les hores de dedicació a cada una de les feines especificades en el diagrama de Gantt. Allà podem veure si cada una de les tasques s'està desviant del temps previst per així poder incidir sobre el problema i en tot cas calcular-ne el cost econòmic i pal·liar-lo amb la partida de contingència.

$\text{Desviació} = (\text{Hores reals} - \text{Hores Estimades}) \cdot \text{preu/h}$

5.2.4 – Desviació respecte al pressupost inicial

Veient els canvis en la planificació, podem dir que no afecta al cost del projecte de manera inesperada, ja que ja teníem l'apartat d'imprevistos, d'on es pagarà la posada en funcionament prevista pel curs vinent.

La posada en funcionament suposarà aproximadament 30h de mà d'obra per explicar com funciona el programa i fer els primers passos amb la persona responsable de l'horari. Per tant tindrà un cost de $30h \cdot 30\text{€/h} = 900\text{€}$ que és inferior al màxim reservat per imprevistos.

5.2.5 – Cost final

Finalment fem balanç dels costos i hem tingut les següents desviacions:

- 900€ extres de la posada en funcionament tal com s'ha explicat en l'apartat anterior que els obtenim del fons de contingència.
- 70€ d'estalvi en la llicència de SublimeText, que finalment no s'ha necessitat.

Part	Cost final
Recursos humans	4.650€ + 900€
Hardware	4.100€
Software	1.520€ - 70€
Despeses indirectes	1.675€
TOTAL	12.775€

El total de 12.775€ és inferior al cost previst a l'inici del projecte (13.745€).

5.3 – Lleis i regulacions

Podem dir que no hi ha cap tema de lleis i regulacions que ens puguin afectar, donat que no estem tractant amb dades privades ni hi ha cap tipus de problema legal en el nostre projecte ja que és d'ús exclusivament intern a l'Escola Súnion i que després no es vendrà ni utilitzarà per a cap altre fi.

5.4 - Coneixements d'assignatures cursades

De les assignatures d'especialitat podríem considerar que m'ajudaran pel projecte les següents assignatures (ordenades de més a menys):

- **IA:** A intel·ligència artificial vam veure algorismes de cerca local durant bona part de l'assignatura. Donat que el problema de resolució d'horaris una possible opció per abordar-lo és amb cerca local, ens serviran els coneixements. També ens servirà l'àmbit de les ontologies per representar el coneixement i les dades associades.
- **A:** Molt a prop de IA, a algorísmia vam fer algorismes greedy, que no deixa de funcionar com la cerca local amb una funció heurística. A part també es van veure molts algorismes famosos importantíssims per conèixer la base de la programació i resolució de problemes NP-complets
- **LI:** Un altre possibilitat per abordar el problema és a través de restriccions lògiques, és a dir transformar el problema dels horaris a un problema SAT per enviar a un SAT solver que el resolgui.
- **APA:** A apa es van treballar sistemes per generar models a partir de dades. Això ens anirà bé per abordar la part desicional del programa quan hi ha incompatibilitats.

- **LP:** A llenguatges de programació vam endinsar-nos en el paradigma de la programació funcional, que posa eines al nostre abast per programar d'una forma, és elegant en certes situacions. Tots aquests coneixements em seran útils de cara a escollir el llenguatge per fer el programa i a l'hora de programar-lo.
- **SID:** A SID principalment es va profunditzar en les ontologies i la representació el coneixement, que ja s'ha comentat que serà d'utilitat.

5.5 - El projecte s'adequa a l'especialitat?

Clarament s'adequa a l'especialitat de computació. Per justificar-ho ens podem recolzar amb la descripció general de l'especialitat a la web:

<< Capacita per dissenyar sistemes informàtics complexos tenint en compte criteris crítics **d'eficiència, fiabilitat**, i seguretat. Prepara per ser capaç **d'avaluar** aquests **requeriments** i **recomanar les màquines**, els **llenguatges de programació** i els **mètodes algorísmics** més adequats per dissenyar-ne una **solució informàtica**. >>

Tal com s'ha comentat el projecte, consisteix a dissenyar una **solució informàtica** tenint en compte **l'eficiència** i la **fiabilitat** per aconseguir que el producte final tingui la funció esperada. Per fer-ho s'hauran de buscar els **mètodes algorísmics** més adequats al problema dels horaris, juntament amb els **llenguatges de programació** idonis per la situació. A més el projecte es du a terme en una empresa, cosa que fa que també es realitzi la posada en producció per tant s'hauran d'evaluar també els **requeriments** per la implementació al servidor definitiu.

5.6 - Competències tècniques

- **CCO1.1:** Avaluar la complexitat computacional d'un problema, conèixer estratègies algorísmiques que puguin dur a la seva resolució, i recomanar, desenvolupar i implementar la que garanteixi el millor rendiment d'acord amb els requisits establerts. **[En profunditat]**

Per resoldre el problema que planteja el projecte, clarament s'ha d'avaluar en profunditat la complexitat computacional del problema i buscar algorismes que garanteixin la resolució correcta del problema, ja que a més tenim el factor important del temps. Per fer-ho haurem d'escollir entre abordar el problema amb una versió SAT o amb una solució basada en heurístiques.

- **CCO1.2:** Demostrar coneixement dels fonaments teòrics dels llenguatges de programació i les tècniques de processament lèxic, sintàctic i semàntic associades, i saber aplicar-les per a la creació, el disseny i el processament de llenguatges. **[Bastant]**

Donat que tot el projecte es basa en la programació d'una solució de software que complexi amb la funció descrita, es demostrarà els coneixements assolits en el marc

dels llenguatges de programació i les tècniques explicades a les assignatures corresponents (herència, polimorfisme, funcions d'ordre superior, etc..).

- **CCO1.3:** Definir, avaluar i seleccionar plataformes de desenvolupament i producció hardware i software per al desenvolupament d'aplicacions i serveis informàtics de diversa complexitat. [**Bastant**]

Donat que s'ha de programar la solució definitiva es buscarà i s'escollirà l'entorn i plataforma més indicats per realitzar la tasca, de la forma més ordenada i que ens ajudi més a mantenir la claredat del programa tot i la complexitat del problema.

- **CCO2.1:** Demostrar coneixement dels fonaments, dels paradigmes i de les tècniques pròpies dels sistemes intel·ligents, i analitzar, dissenyar i construir sistemes, serveis i aplicacions informàtiques que utilitzin aquestes tècniques en qualsevol àmbit d'aplicació. [**En profunditat**]

El projecte principalment es fonamenta en una intel·ligència artificial que sigui capaç de generar horaris a partir d'unes certes restriccions i prendre desicions en el cas d'incompatibilitats de factors. Per tant s'haurà de treballar el paradigma dels sistemes intel·ligents.

- **CCO2.2:** Capacitat per a adquirir, obtenir, formalitzar i representar el coneixement humà d'una forma computable per a la resolució de problemes mitjançant un sistema informàtic en qualsevol àmbit d'aplicació, particularment en els que estan relacionats amb aspectes de computació, percepció i actuació en ambients o entorns intel·ligents. [**En profunditat**]

S'haurà de representar el coneixement i dades relacionades amb l'horari primerament dissenyant una ontologia per posteriorment passar-ho a una base de dades on puguem treballar fàcilment amb les dades.

- **CCO2.3:** Desenvolupar i avaluar sistemes interactius i de presentació d'informació complexa, i la seva aplicació a la resolució de problemes de disseny d'interacció persona computador. [**Una mica**]

Donat que una part del projecte consisteix a construir la interfície d'entrada de restriccions, s'haurà de dissenyar una interfície que simplifiqui al màxim l'entrada d'informació complexa de manera que després el programa pugui abstroure i processar.

- **CCO2.4:** Demostrar coneixement i desenvolupar tècniques d'aprenentatge computacional; dissenyar i implementar aplicacions i sistemes que les utilitzin, incloent les que es dediquen a l'extracció automàtica d'informació i coneixement a partir de grans volums de dades. **[En profunditat]**

Per fer el sistema de decisió ens basem en una base de dades d'horaris fets per un humà durant els últims anys, d'aquesta manera podrem generar uns models decisionals a partir d'aquestes dades i per tant aprenentatge computacional.

- **CCO3.1:** Implementar codi crític seguint criteris de temps d'execució, eficiència i seguretat. **[Bastant]**

Tal com s'ha comentat el criteri del temps i per tant de l'eficiència són crítics, per tant es treballarà la competència.

- **CCO3.2:** Programar considerant l'arquitectura hardware, tant en ensamblador com en alt nivell. **[Una mica]**

Donat que s'ha de fer la implementació final del producte es poden fer millores de codi en alt nivell pensant en la cache de la màquina final i de tota l'arquitectura en general.

6 - Sostenibilitat

6.1 – Matriu de sostenibilitat

A partir de les preguntes i respostes puntuem els valors de cada casella dins del rang marcat. Com veiem hi ha un risc ambiental durant tot el projecte ja que hi haurà sempre un consum energètic. En canvi en apartat econòmic i social un cop passats la fase de desenvolupament les puntuacions pugen.

	PPP	Vida útil	Riscos	TOTAL
Ambiental	7	15	-1	22
Econòmic	7	19	-1	26
Social	9	20	0	29
TOTAL	25	54	-2	77

Per intentar pal·liar els punts negatius de la fase de funcionament, no hi ha cap possible sortida viable, donat que els costos mediambientals de les possibles alternatives superarien els riscos actuals.

6.2 – Dimensió Econòmica

Existeix una avaluació de costos, tant de recursos materials com humans? Sí, existeix en els dos àmbits.

S'ha tingut en compte el cost dels ajustaments / actualitzacions / reparacions durant la vida útil del projecte? No s'han tingut en compte ja que s'ha fet l'anàlisi de la part econòmica de la realització del projecte.

El cost del projecte ho faria viable si hagués de ser competitiu? Sí ja que no hi ha cap altre producte que compleixi totes les especificacions. Per altra banda hi ha el problema que tampoc hi ha gaire demanda d'un producte tan complert.

Es podria realitzar un projecte similar en molt menys temps o amb molts menys recursos i, per tant, menor cost? No, el projecte ja està amb el mínim de recursos, en tot cas es podria intentar reduir despeses de software fent-ne servir més open source.

El temps dedicat a cada tasca és proporcional a la seva importància (s'ha dedicat molt de temps a desenvolupar parts del projecte que podien haver estat reutilitzades de tecnologies / projectes / coneixements existents)? Sí tot té una certa proporcionalitat.

Està prevista o hi ha col·laboració amb algun altre projecte (acadèmic, empresa, associació, etc.)? No

6.3 – Dimensió Social

Quina és la situació social i política del país / lloc / ciutat / ... on realitzaràs el teu projecte? I la del sector a què inclou el teu projecte? La situació política és inestable amb un canvi de govern en curs, en el sector concret de l'educació és un moment de canvis.

Creus que la teva activitat podria afavorir o empitjorar aquesta situació? La situació global del país òbviament no, però la comoditat i efectivitat de la generació d'horaris a Súnion clarament millorarà.

Hi ha una necessitat real del teu producte / servei? Sí, ja que l'encarregat de fer l'horari els últims 45 anys en breu es jubilarà i aquesta és una forma de simplificar la tasca, mantenint l'horari que forma part de la identitat del centre escolar.

Satisfereix aquesta necessitat millora la qualitat de vida dels consumidors? En part sí, ja que no hauran de perseguir a una persona per demanar les seves necessitats d'horari, ho podran fer a través d'una web a qualsevol hora.

El resultat del projecte, En què / Com canviarà la vida de l'usuari? L'usuari podrà canviar i demanar les seves necessitats des de qualsevol lloc a qualsevol hora.

6.4 – Dimensió Ambiental

Quins recursos es necessitaran en les diferents fases del projecte? Principalment tot el projecte tindrà un impacte mediambiental amb l'electricitat i la petjada ecològica del hardware fet servir.

Quin consum tindran aquests recursos durant el desenvolupament del projecte i posteriorment durant la seva posada en marxa i vida útil? Quin és l'impacte ambiental d'aquest consum? Depèn del consum elèctric que tingui el servidor on s'allotjarà el software i la companyia elèctrica quin sistema/es de generació d'electricitat faci servir.

Quin consum i impacte ambiental tindria realitzar la mateixa activitat sense l'existència del teu TFG (estalvi de paper i altres materials i/o energia?) Es fa servir molta més energia ja que hi ha una infraestructura d'un ordinador amb dues pantalles tot el dia encès per poder dissenyar el horari, a més s'estalviarà paper que es fa servir per anar apuntant tot el que els professors demanen.

Quins recursos poden reaprofitar d'altres projectes? Es pot reaprofitar el servidor, ja que hi ha algun que a final de curs quedarà en desús.

Durant el desenvolupament del teu producte es generarà algun tipus de contaminació? Només la relacionada amb la generació d'electricitat.

Requereix el projecte de material manufacturat i en quina quantitat? És ètic el seu origen i desenvolupament? Ho és la seva empresa fabricant o distribuïdora? Es necessita el servidor i l'ordinador que el dos són de marca HP que és un fabricant amb una normativa mediambiental.

Amb la implantació del projecte s'augmenta o es disminueix la petjada ecològica? Es disminueix, ja que no es gasta paper i es redueix el consum elèctric.

Quines parts del projecte podran reciclar-se o reutilitzar-se en altres projectes? El hardware es pot reutilitzar, així com el software utilitzat pel desenvolupament.

7 – Conclusions i futur

Podem concloure que el treball ha arribat a bon port, ja que s'han acomplert els objectius de crear un programa de generació d'horaris personalitzat per a l'Escola Súnion i fer-ne la implantació.

S'ha escollit de manera efectiva el procediment per abordar la resolució del problema a partir de certes possibilitats i tenint sempre en compte els requisits que s'havien determinat en fer l'anàlisi del projecte.

De les restriccions que s'havien previst inicialment s'han implementat totes d'una manera o altra, tot i que ha faltat per implementar el fet de poder escollir un tipus d'aula concreta (multimèdia, ordinadors, pissarra...), és a dir poder escollir les aules a traves de conjunts segons l'equipament del que disposen. De tota manera com que es poden escollir aules concretes, es pot arribar al mateix resultat posant manualment totes aquelles que ens interessin.

Un dels fronts que ha quedat obert és resoldre els canvis d'última hora de l'horari. S'ha decidit fer-los a mà, ja que és molt difícil decidir amb el programa quins canvis s'han de fer per afectar, en la menor mesura, l'organització de l'escola quan és un canvi a tan curt termini.

Tal com s'havia previst s'ha deixat el programa preparat per fer-li millores i afegir-li les extensions que es vulgui de manera mes o menys senzilla. D'aquesta manera en un futur es podran afegir restriccions no contemplades en el disseny del TFG.

Recordem que tant la interfície com el generador estan programats amb classes i per tant són molt modulables, és a dir que no tindrem problemes per fer extensions.

8 - Bibliografia

8.1 - Referències

- 1) Russell, S., & Norvig, P. (2009). *Artificial intelligence: a modern approach*. Prentice Hall.
- 2) Béjar, J. (2011). *Universitat Politecnica de Catalunya*. Recollit de http://www.cs.upc.edu/~bejar/ia/transpas/teoria/2-BH3-Busqueda_local.pdf
- 3) Berzal, F. (sense data). *Universidad de Granada*. Recollit de Planificación: <http://elvex.ugr.es/decsai/iaio/slides/P1%20Planning.pdf>
- 4) *Nevonprojects*. (Abril / 2015). Recollit de Automated College Timetable Generator: <http://nevonprojects.com/automated-college-timetable-generator/>

8.2 - Altres

Béjar, J. (2011). *Universitat Politecnica de Catalunya*. Recollit de http://www.cs.upc.edu/~bejar/ia/transpas/teoria/2-BH3-Busqueda_local.pdf

Berzal, F. (sense data). *Universidad de Granada*. Recollit de Planificación: <http://elvex.ugr.es/decsai/iaio/slides/P1%20Planning.pdf>

Generic Timetable Generator. (Abril / 2013). Recollit de <https://sourceforge.net/projects/generictimetabl/>

Nevonprojects. (Abril / 2015). Recollit de Automated College Timetable Generator: <http://nevonprojects.com/automated-college-timetable-generator/>

Russell, S., & Norvig, P. (2009). *Artificial intelligence: a modern approach*. Prentice Hall.

Timetable Generator. (Agost / 2012). Recollit de <https://sourceforge.net/projects/ttgen/>

Wikipedia. (Novembre / 2015). Recollit de Programación con restricciones: https://es.wikipedia.org/wiki/Programaci%C3%B3n_con_restricciones

9 – Annex

9.1 – Manual de l'usuari

En aquest manual s'explica el funcionament de la interfície d'entrada de dades pel generador d'horaris, separat per pàgines. A totes les pàgines trobarem els botons d'afegir, editar i eliminar, on el botó d'editar té exactament el mateix funcionament que el d'afegir però editant les dades.

Conceptes previs:

Durant tot el manual es fa referència a conceptes bàsics de l'horari de Súnion que cal saber:

- Sessió/Targeta: És una activitat dins de l'horari, que té com a elements propis la duració, les aules preferides per fer aquesta activitat, els professors que la realitzen i per descomptat els grups que la realitzen.
- Grup: És l'agrupació més petita d'alumnes que hi ha a l'escola, són grups de 8 persones que formen l'anomenat grup o grup natural.
- Duració: Cada targeta com ja hem dit té una duració associada, que ve determinada en segments, és a dir en fraccions de 15 minuts.
- Segment: Fraccions de 15 minuts que determinen les hores d'inici o finalització d'una activitat. El matí conté 20 segments i la tarda 9.

Dades predefinides:

En aquest apartat de la interfície podem determinar les sessions habituals i les restriccions horàries que es repeteixen setmanalment.

- Sessions: fem clic al botó d'afegir i se'ns obre un assistent per crear una sessió on ens demanarà les dades bàssies:
 - o Curs
 - o Assignatura
 - o Grup inicial i nombre de grups consecutius que han de fer l'activitat.
 - o Duració
 - o En cas que cada setmana aquesta activitat es faci en el mateix dia i hora, s'han d'entrar les dades de dia fixat i segment fixat.
 - o Professors responsables
 - o Aules preferides
- Restriccions horàries: fem clic al botó d'afegir i se'ns obre un assistent per crear una restricció amb les següents dades:
 - o Nom del professor
 - o Dia de la setmana
 - o Hora d'inici i hora de finalització

Setmanes:

En aquest apartat podem definir les setmanes que volem poder gestionar. La setmana és l'objecte que podem enviar al generador perquè ens prepari una solució. Per crear-ne una farem clic al botó d'afegir i omplirem les següents dades:

- Nom identificatiu de la setmana (ex: 27 de juny)
- Cada dia de la setmana l'hem d'associar a un dia concret del calendari, si el deixem buit és que aquell dia és festiu.

Dades:

En aquest apartat quan entrem ens demana seleccionar una setmana. Un cop seleccionada el funcionament és idèntic al de l'apartat de dades predefinides.

Podem observar com al entrar en una setmana nova, ja hi ha dades, ja que el sistema ha copiat les dades predefinides a la setmana creada, de manera que editem "la normalitat" per poder decidir totes les dades d'aquella setmana.

Execucions:

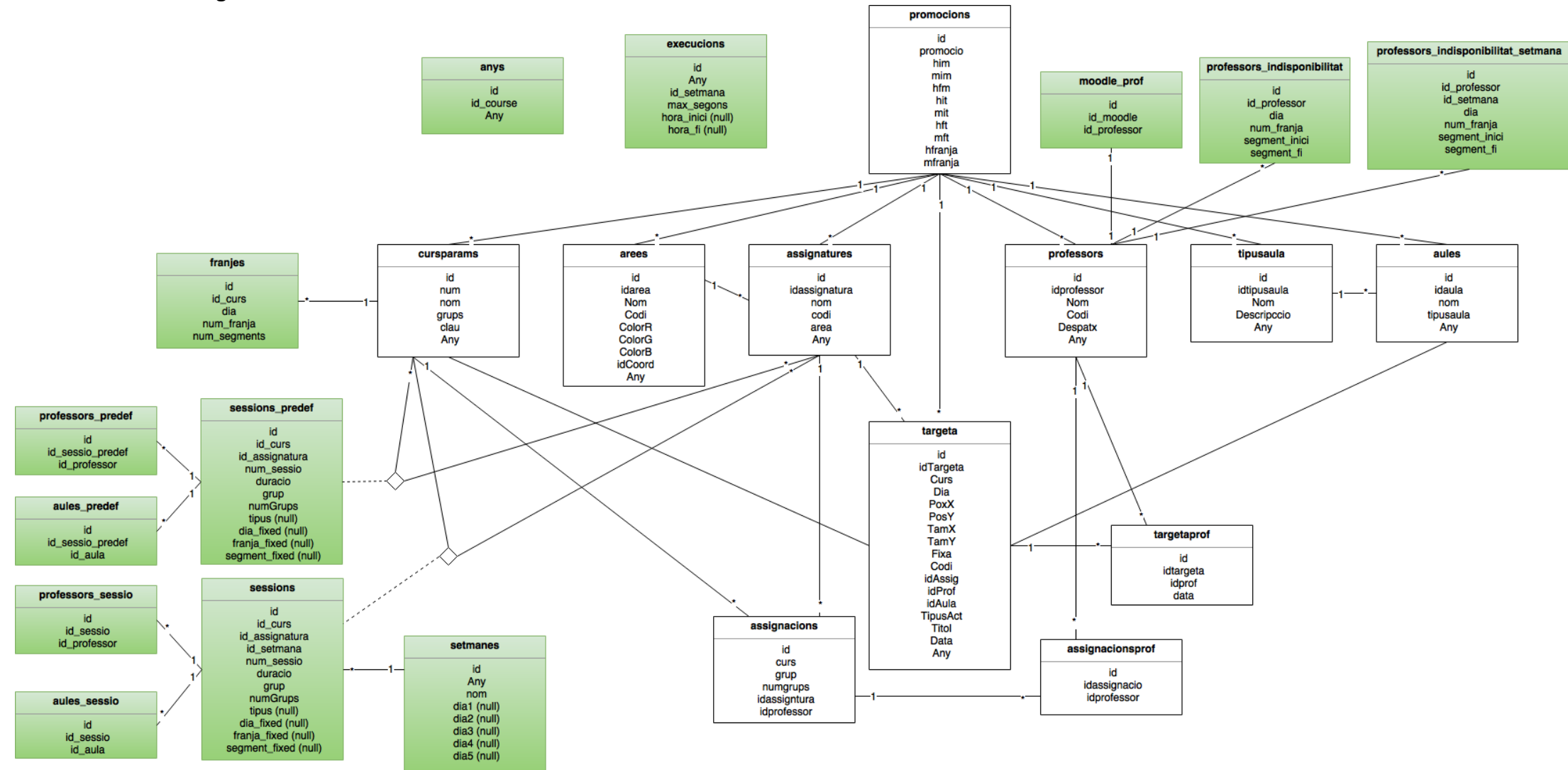
En aquest apartat podem definir les execucions del generador d'horaris i descarregar-ne els resultats.

Per fer una nova execució fem clic al botó d'afegir i omplim les dades necessàries:

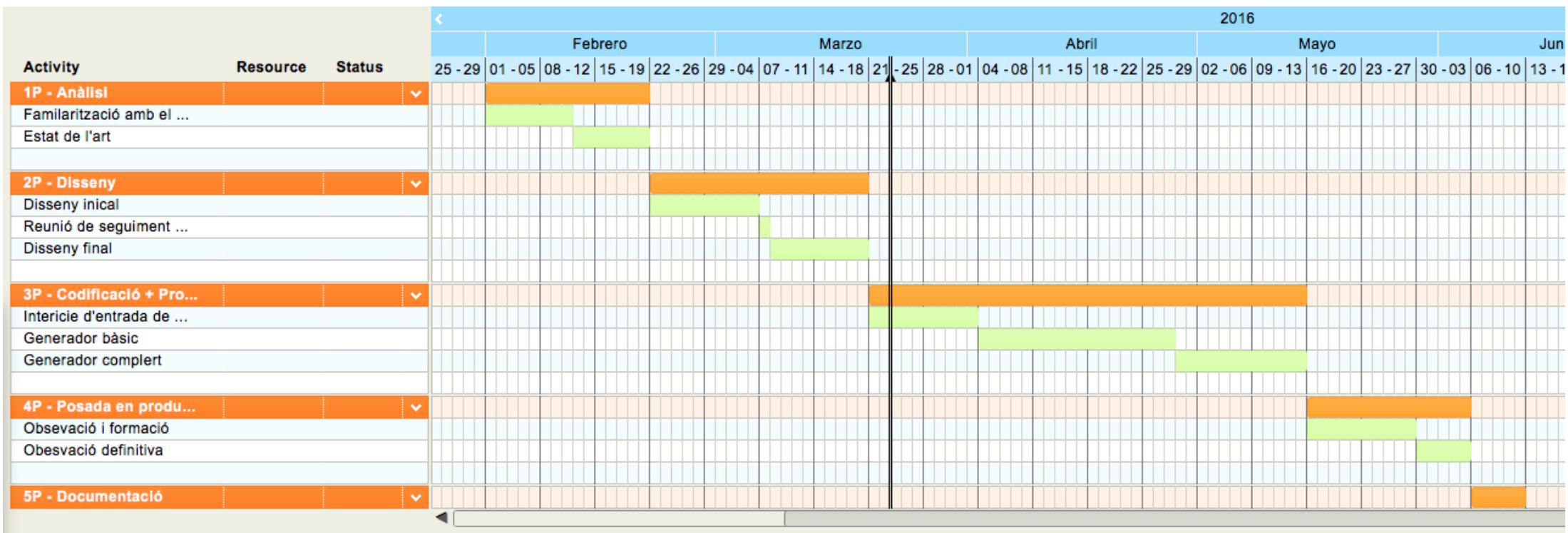
- Setmana de la qual es vol fer el càlcul
- Nombre màxim de segons que volem que el generador dediqui a buscar una solució.

Un cop creada una execució, hem d'esperar que acabi per poder descarregar-nos el resultat fent clic al botó de descarregar.

9.1 – Diagrama Base de Dades



9.1 – Gantt planificació inicial



9.2 – Gantt planificació modificada

